IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the application of	:	**CROY, Timothy et al.**
Serial No.	:	**10/672,900**
Filed	:	**September 26, 2003**
For	:	**Management of Network Elements Using a Proxy Agent**
Examiner	:	**JEAN, Frantz B.**
Art Unit	:	**2154**
Customer No.	:	**23644**

BRIEF ON APPEAL

Honorable Director of Patents and Trademarks
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This Appeal is from the Examiner's final Office Action dated September 6, 2007 in which claims 1, 2, and 4-19 of this application were finally rejected. A Notice of Appeal was filed December 6, 2007.

The fee of $510.00 pursuant to 37 C.F.R. §41.20(b)(2) for this brief should be deducted from Deposit Account No. 12-0913.

(i) REAL PARTY IN INTEREST

The Assignee, Nortel Networks Limited, is the real party in interest in the pending appeal.

(ii) RELATED APPEALS AND INTERFERENCES

Applicants are unaware of any other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii) STATUS OF CLAIMS

The application was filed with Claims 1-19, and Claim 3 was later cancelled. Claims 1, 2, and 4-19 are pending in the Application, are finally rejected, and are the claims that are being appealed. Claims 1, 2, and 4-19 are set forth in the Claims Appendix.

(iv) STATUS OF AMENDMENTS

No claim amendments have been filed subsequent to the final rejection dated September 6, 2007. A response without amendments was filed November 6, 2007 and was not entered for inexplicable reasons.

(v) SUMMARY OF CLAIMED SUBJECT MATTER

The invention relates generally to communication between systems which support incompatible protocols using a proxy agent.

Independent claim 1 relates to a proxy agent (150 in Fig. 4) for communicating data components (42 in Fig. 3; 158 in Figs. 5A and 5B) between a first system (112 in Fig. 4) which supports a first protocol and a second system (114 in Fig. 4) which supports a second protocol, said first and second protocols being mutually incompatible (see page 6, lines 8-19), the proxy agent comprising a directory (152 in Fig. 4) for storing said data components, said directory supporting a hierarchical data structure (Figs. 5A and 5B, and see page 12, lines 4-21), wherein each data component stored in said directory is associated with a first data component identifier (160 in Figs. 5A and 5B) which is compatible with said first protocol, and with a second data component identifier (162 in Figs. 5A and 5B) which is compatible with said second protocol,

and wherein each stored data component is associated with a respective position in the hierarchical data structure (see page 12, lines 23-28).

Briefly, therefore, the method of claim 1 relies on a proxy which includes a directory supporting a hierarchical data structure of data components, each such component having two identifiers, so that one identifier ensures compatibility with two mutually incompatible protocols.

Independent claim 17 relates to a network comprising a first system (112 in Fig. 4) which supports a first protocol and a second system (114 in Fig. 4) which supports a second protocol, said first and second protocols being mutually incompatible (see page 6, lines 8-19), and a proxy agent as claimed in Claim 1.

Independent claim 18 relates to a method of communicating data components between said first system (112 in Fig. 4) which supports a first protocol and a second system (114 in Fig. 4) which supports a second protocol, said first and second protocols being mutually incompatible (see page 6, lines 8-19), said method comprising storing said data components (42 in Fig. 3; 158 in Figs. 5A and 5B) in a directory (152 in Fig. 4) wherein each data component stored in said directory is associated with a first data component identifier (160 in Figs. 5A and 5B) which is compatible with said first protocol, and with a second data component identifier (162 in Figs. 5A and 5B) which is compatible with said second protocol.

(vi) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

There is one ground of rejection to be reviewed:

1) Whether claims 1, 2, and 4-19 are unpatentable under 35 U.S.C. 102(e) as being anticipated by Page et al. US Patent No. 7,024,476.

(vii) ARGUMENT

Rejection under 35 U.S.C. 102(e) over Page et al.

Independent Claims 1, 17 and 18

Both the claimed invention and the prior art reference are concerned with communications between two systems one of which uses a first protocol and the other of which uses a second, incompatible protocol.

In the described embodiments, the claimed invention is illustrated with reference to a management or directory server and a plurality of managed devices on the same network, the managed devices and the directory server employing different and mutually incompatible protocols. Page's described system is of the same general type.

To avoid any confusion in terminology, between that used by Page and that used in the instant application, the following common terminology will be used for the main components of such architectures, namely the *directory server*, the *managed devices*, and the *proxy*.

"Directory Server": This is the server which stores and maintains the settings and capabilities of the managed devices under its management. In Page (Figs. 2 and 5), this is the "directory server 25". In the present application, the directory server is implemented as a network management system (NMS) 12 and its associated database or management information base (MIB) 13, as described at page 8, lines 22-25.

"Managed Devices": These are items such as printers, routers, etc. which are connected to the network and for which the directory server stores settings and capabilities. By issuing appropriate commands and queries, the directory server can manage and control such devices (with the assistance of the proxy as an intermediary). Page shows these examples of such managed devices as "SNMP

device 30 (16,17)" and "Hybrid device with embedded LDAP client and SNMP 31(15)" in Figs. 2 and 5. The present application refers to them as "Non SNMP network element 114" in Fig. 4.

"Proxy": The proxy translates between the protocol used by the directory server and the protocol used by the managed devices. Page refers to this as the "directory proxy 29", while the present application refers to it as the "proxy agent" or simply "proxy 150". This is the claimed "proxy agent" of claim 1, of course.

With those terms in mind, Applicants submit that:

(i) Page's "hierarchical directory" is located in a directory server connected to a proxy, and therefore **not** in the proxy itself as required by the claims under examination; and

(ii) Page's proxy is of the general type shown in Fig.1 of the present application, which Applicants describe as a conventional proxy, i.e. it has a translator which is ignorant of any hierarchical directory structure, and therefore suffers from the shortcomings described in relation to the system of Fig. 1 of the present application.

In the following discussion, references to a hierarchical directory, and to the fact that such a directory is lacking from Page's proxy, should be taken as a shorthand reference to:

- a directory for storing data components
- the directory supporting a hierarchical data structure
- each stored data component being associated with a respective position in the hierarchical data structure.

1) Page's hierarchical directory is part of the directory server, not the proxy

Applicants have consistently argued that Page's proxy had no hierarchical directory. This argument was not directly rebutted in the final Office Action, but the wording of the rejection was updated to include a specific reference to column 2, lines 38-59, where such a hierarchical directory could allegedly be found in Page et al.

However, this passage clearly describes a directory server, not a proxy. It simply describes how such a directory server organises its descriptions of device settings and capabilities in a hierarchical manner. An example is given of how each managed device finds its place within the structure, e.g. the directory branch for network printers has a sub-branch for ink jet printers, which in turn has a plurality of entries for storing the settings and capabilities corresponding to each of the ink jet printers on the network. The passage is of no relevance to a proxy in such a managed system.

Claim 1 of the instant application is concerned only with the proxy (or "proxy agent"). The entire argument in the Office Action's rejection stands or falls on whether Page discloses a proxy having the required directory with a hierarchical data structure. Because the passage relied on does not fulfil this heavy burden imposed on it, the rejection fails on a fundamental point – Page contains no description of a proxy with a hierarchical directory as claimed in claim 1, and Page only foresees the directory server itself as having such a hierarchical directory. Accordingly, the claimed invention is not anticipated by Page for this reason.

2) Page's proxy is blind to directory structure

The conclusion above is determinative of the issue of anticipation, but Applicants wish to make a further point. The description of how Page's proxy operates leaves no doubt that it simply translates from one protocol to another blindly, and without any understanding of or awareness of a hierarchical directory structure according to which the managed devices may be organized.

Firstly, one can compare Fig. 1 of the present application with Fig. 5 of Page et al. In each case there is a directory server communicating with a first protocol handler or client, and there are managed devices communicating with a second protocol handler or client. The protocol handlers form part of the respective proxies, and translation is provided between the two protocols in each proxy.

Page uses the "LDAP/SNMP translator 64" for this translation, while the present application uses the "mapping component 22" with the aid of the "mapping definition 24" and "cache 26". Both translation facilities operate the same, however: they

simply translate commands, queries and responses from one protocol into the corresponding commands, queries and responses of the other protocol.

The fact that Page's translator 64 operates blindly and without any knowledge of the directory structure maintained by the directory server, can be seen from the following description of operation at column 15, lines 18-31, which describe the operation of the translator 64 when a new device has been located on the network.

> *Translator 64 formats the device's information into LDAP format, communicates with LDAP client 60 and sends the LDAP formatted SNMP device's information to LDAP client 60 (step S707). LDAP client 60 then establishes communication with directory server 25 to self publish the SNMP device's information to the directory server (step S708). LDAP client 60 first utilizes an LDAP_ADD command to attempt to add the SNMP device's information in directory server 25. If an entry for the SNMP device is already present in directory server 25, then an error message is returned by the directory server to LDAP client 60. LDAP client 60 then utilizes an LDAP_MODIFY command to replace the directory entry information in the directory entry of directory server 25 for the existing device.*

The translator 64 and the LDAP client 60 are clearly working with no knowledge of the directory structure employed by the directory server, since they simply attempt to add information without knowing whether the device is already in the directory server's hierarchy or not. If the LDAP_ADD procedure succeeds, then the information is added, while if it fails, the client is programmed to retry using a MODIFY command. If the proxy's translator or the proxy's client 60 had a copy of the hierarchical directory maintained on the directory server, then this trial-and-error update procedure would make no sense. The same trial-and-error attempt at adding and then modifying directory entries, incidentally, is repeated throughout the description.

Conclusions

In conclusion, the proxy of Page et al. has no hierarchical directory as required by claim 1. Furthermore, the description of how the Page et al. proxy operates

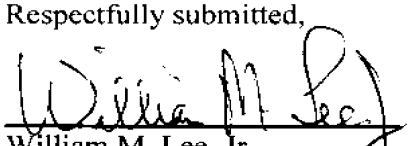
demonstrates that it has no hierarchical directory capabilities or knowledge, and simply translates commands from one format to another without employing the hierarchical directory features set out in claim 1.

The arguments made above in relation to claim 1 apply equally to independent claims 17 and 18, and to each of the dependent claims which share at a minimum the features of the independent claims from which they depend.

Therefore, it is submitted that the Examiner's rejections are clearly in error, and should be reversed.

February 6, 2008

Respectfully submitted,



William M. Lee, Jr.
Registration No. 26,935
Barnes & Thornburg LLP
P.O. Box 2786
Chicago, Illinois 60690-2786
(312) 214-4800
Fax (312) 759-5646

CLAIMS APPENDIX

1. A proxy agent for communicating data components between a first system which supports a first protocol and a second system which supports a second protocol, said first and second protocols being mutually incompatible, the proxy agent comprising a directory for storing said data components, said directory supporting a hierarchical data structure, wherein each data component stored in said directory is associated with a first data component identifier which is compatible with said first protocol, and with a second data component identifier which is compatible with said second protocol, and wherein each stored data component is associated with a respective position in the hierarchical data structure.

2. A proxy agent as claimed in Claim 1, further including a first protocol handler arranged to communicate with said first system using said first protocol, and a second protocol handler arranged to communicate with said second system using said second protocol, wherein said first protocol handler is arranged to send data components to, and/or receive data components from, said directory using said first data component identifier, and said second protocol handler is arranged to send data components to, and/or receive data components from, said directory using said second data component identifier.

3. (Cancelled)

4. A proxy agent as claimed in Claim 1, wherein said respective first data component identifiers support a hierarchical structure and serve to identify the respective position of the respective data component in the hierarchical data structure.

5. A proxy agent as claimed in Claim 4, wherein said data components are arranged into directory entities within the directory, each directory entity comprising a one or more directory entries, each directory entry comprising a respective data component, a respective first data component identifier and a respective second data component identifier.

6. A proxy agent as claimed in Claim 5, wherein each data component within a directory entity belongs to the same branch of the hierarchical data structure.

7. A proxy agent as claimed in Claim 5, wherein each directory entity is associated with a first directory entity identifier which is compatible with said first protocol and with a second identifier which is compatible with said second protocol.

8. A proxy agent as claimed in Claim 7, in which said respective first directory entity identifiers support a hierarchical structure.

9. A proxy agent as claimed in Claim 7, in which each of said first directory entity identifiers belongs to a branch of the hierarchical data structure that is one hierarchical level above the branch to which the respective data components in the respective directory entity belong.

10. A proxy agent as claimed in Claim 5, in which a respective schema is provided to define each type of directory entity and wherein a respective directory entry is created by populating a respective schema with one or more data components.

11. A proxy agent as claimed in Claim 1, wherein said first protocol supports a hierarchical data structure.

12. A proxy agent as claimed in Claim 1, wherein said first and second protocol each comprise a respective network management protocol.

13. A proxy agent as claimed in Claim 1, wherein said first protocol comprises Simple Network Management Protocol (SNMP).

14. A proxy agent as claimed in Claim 1, wherein said first system comprises a Network Management system (NMS) and said second system comprises a network element.

15. A proxy agent as claimed in Claim 14, wherein said proxy agent effects communication between said Network Management system and a plurality of network elements, at least some of said network elements supporting said second protocol.

16. A proxy agent as claimed in Claim 1, wherein said directory comprises a directory which supports Lightweight Directory Access Protocol (LDAP).

17. A network comprising a first system which supports a first protocol and a second system which supports a second protocol, said first and second protocols being mutually incompatible, and a proxy agent as claimed in Claim 1.

18. A method of communicating data components between said first system which supports a first protocol and a second system which supports a second protocol, said first and second protocols being mutually incompatible, said method comprising storing said data components in a directory wherein each data component stored in said directory is associated with a first data component identifier which is compatible with said first protocol, and with a second data component identifier which is compatible with said second protocol.

19. A computer program product comprising machine-readable instructions which when executed in a computer cause said computer to perform the method of claim 18.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None